UNIVERSITY OF GOTHENBURG

Visualizing the status of telecommunication networks with dashboards

Bachelor of Science Thesis in Software Engineering and Management

MAGNUS HERNEGREN
NIKLAS LARSSON

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
Göteborg, Sweden, June 2015

Visualizing the status of telecommunication networks with dashboards

MAGNUS HERNEGREN
NIKLAS LARSSON

Examiner: MORGAN ERICSSON

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden June 2015

Visualizing the status of telecommunication networks with dashboards

Magnus Hernegren
Software Engineering and Management
University of Gothenburg
gusaheqama@student.gu.se

Niklas Larsson
Software Engineering and Management
University of Gothenburg
guslarnih@student.gu.se

Abstract

Background: KPIs/PIs (Key Performance Indicators/Performance Indicators) are types of indicators that are of importance to stakeholders in order for the strategic goals of the organisation to be met. When visualizing datasets, displaying right KPI/PI for the right stakeholder is important, otherwise he/she cannot act on the given information.

Purpose: To develop and evaluate the effectiveness of using dashboards as a visualization tool for KPI/PI values for a telecommunication network.

Research Method: Design research was used for this thesis. To validate the results two prototypes of visualization solutions were developed and interviews were conducted to evaluate the usefulness of the visualization solutions.

Results: The research yielded in six dashboards visualizing the status of a telecommunication network, three of which visualized a test network, and three of which visualized real data from the field.

Conclusions: It is an appropriate solution to visualize KPI/PI values using dashboards. Dashboards could be made more useful in certain situations by introducing drilldown features. Developing dashboards using Dashing-js is a quick and efficient way to visualize the status of telecommunication networks.

1 Introduction

Visualization of data has a long history in Ericsson, from the first spreadsheets which displayed the data as numbers (Johansson, Meding, & Staron, 2007) to the more modern gadget-based interfaces (Staron, Meding, & Nilsson, 2009) (Staron, Meding, & Palm, 2012) (Shollo, Pandazo, Staron, & Meding, 2010). The company is transitioning from the current phase of gadgets to dashboard-based information presentation. This is in correlation to a more generic way of working, which enables teams to create their own independent dashboards, to monitor their specific areas of interest.

Currently there is a need to visualize the status of a test telecommunication network at Ericsson. The reason for this is to be able to validate their solutions, innovations, and experiments in a customer-like environment, thus shortening the feedback-loop considerably, e.g. while testing features on the network.

The *status* of this network that needs to be visualized in this case is a broad term, but can be reduced to the concept of *data regarding current activities in the network*. These data can for example be

- The number of users that are currently using the network.
- How much bandwidth that is available per user / in total.
- How much CPU load the current activities are applying on a node.
- How many failed connections that have occurred or how much data that has been dropped due to errors.

Dashboards are one way to visualize the status of a network. It functions as an overview of measurements, providing a simple summary. The company works according to lean and agile paradigm, it has agile teams that have a holistic responsibility for the features they develop and test. Some of the teams started to develop dashboards by themselves in order to get an overview of the features that they were responsible for. It became the task of the metrics team, a team in the company concerned with measurements and developing in-house visualization products, to turn these dashboards into a generic product for all teams to use and benefit from. They also had the challenge of creating a visualization of the test network, which would give other teams than the test team the possibility to view the status of it. Although in order for the metrics team to develop and maintain a lager set of dashboards; a need for a generic solution became apparent.

The purpose of this study was to research and develop a set of dashboards to visualize the status of the test network. Using a design research method, the researchers collected information and analysed the present situation in the company regarding challenges with visualization of metrics. By identifying the visualization needs the researchers could develop a solution in the form of dashboards. The main research question is as follows: *How to visualize the status of a telecommunication network using dashboards?*

The result of this research consists of a number of different dashboards, whose content is grouped by their type of data. This solution resolves the organizational need of visualizing the status of a telecommunication network. It also consists of a prototype demonstrating how data on a dashboard can be manipulated using interactive features on a web page. The research also provides information on how a dashboard is built up from different components to facilitate further maintenance and development of dashboards. The research was evaluated by conducting interviews. The interviews proved that the developed dashboards satisfy much of the visualization need in the company, but that there also is a desire for additional functionality. The validation also shows that data such as *ratios* and *numbers* easily and effectively can be visualized and communicated using dashboards.

2 Background

This section describes terms and techniques used in the thesis. The purpose of this section is to give definitions and motivation behind design choices when developing visualization products.

2.1 Visualization

Williams, Sochats, and Morses (1995) definition of visualization is that it's a "*cognitive process performed by humans in forming a mental image of a domain space. In computer and information science it is, more specifically, the visual representation of a domain space using graphics, images,* *animated sequences, and sound augmentation to present the data, structure, and dynamic behaviour of large, complex datasets that represent systems, events, processes, objects, and concepts"*.

According to Telea (2014), data visualization has the purpose of providing insight by using graphical representations of data. The insight it provides is created in order to answer specific questions, or to allow people to explore the presented facts and draw conclusions on their own (see Figure 1). In the first case, the types of specific questions can be divided into categories of quantitative and qualitative questions. Whereas visualizations for quantitative questions most commonly display a single number in a simple way (i.e. a max/min value to answer questions like "What is the max value of y?"), visualizations for a qualitative question can for example show values over time as a line graph (e.g. to answer a question like "Where was the X value above the threshold?"). Exploratory visualizations on the other hand can be created without a specific question in mind, but are rather created to get a view of a process. They are relevant in contexts where the visualization of the data is *believed* to provide useful information, but when it cannot be determined beforehand what questions it will answer, or if it will be of relevance at all.

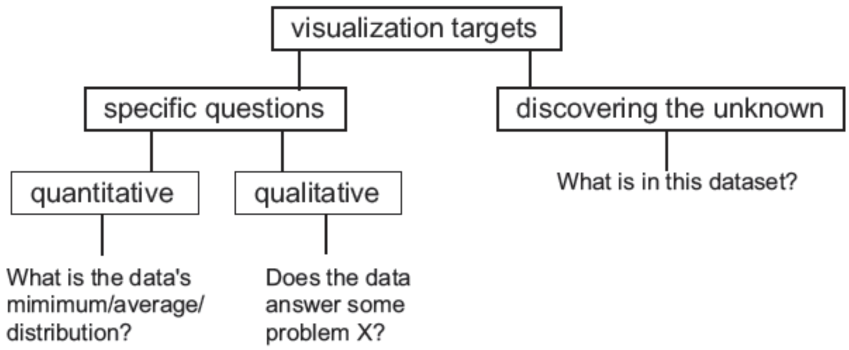

Figure 1: Different kinds of questions that visualization can answer (Telea, 2014)

This thesis uses theories on correct visualization in order to be able to monitor the status of the network, and from there be able to observe links between issues. Staron et al. (2013) have conducted research relating to the importance and effects of correct visualization of data. This research was done in relation to visualizing the stability of a product during development. The research showed that being able to use this visualized stability data during development increased efficiency in the development process. Staron and Meding (2011) have researched how to monitor bottlenecks in a project using indicators in a measurement system. The research investigated how and what data was relevant to the involved developers. They developed a tool that presented the data that their research proved was relevant to the developers. The visualized data in the tool proved to be effective in identifying and anticipating problems in the project. Burkhard et al. (2005) present questions that need to be answered regarding the goal of the visualization, the knowledge to be visualized, the recipient of the visualization, and the method of how to visualize it, in order for a visualization to be effective (Burkhard, Spescha, & Meier, 2005). By answering these questions this research could develop dashboards that were satisfied the requirements of the stakeholders.

2.2 Dashboards

Few (2006) defines a dashboard *as an easily interpreted display of information that is needed to achieve an objective*, and explains that dashboards as a tool for visualization of data can be very powerful. It highlights the importance of showing the right data in the right way, and points out a lot of pitfalls that are common when designing dashboards. It explains that access to data is not in itself useful, if it cannot be presented in a comprehensible way.

Dashboards (see Figure 2) are platforms for creating visualizations to answer both specific questions and to allow people to explore data. A dashboard consists of one or several widgets that can display whatever information, and they can also be used as a foundation for collecting a number of data indicators and graphs to provide a graphical representation of these on a screen. By tailoring the dashboards content according to needed information, or to explore certain data, it's possible to create a customized view to satisfy these needs. For example it can provide a number of indicators showing the immediate status of some part of the data (e.g. to answer a quantitative question such as "what is the current/max/min level of this data from x"), it can also provide answers to qualitative questions in the form of line graphs or other visualizations that display data over time (e.g. "how has x developed over the last month"). An exploratory aspect of this can be to correlate the different data views and draw conclusions based on them.

Dashboards which are easily configurable to display data that is relevant to a specific team will fill a gap which desktop gadgets can't fill; for example the need for more information and higher detail in the graphical visualization. Dashboards should be able to provide an easily understood overview of relevant data in real-time which will enable the team to get feedback instantly on their performance or on their products. Dashboards should also allow the teams to get a compiled view of the indicators that are necessary. In this research, dashboards are used to answer quantitative and qualitative questions, as well as have the potential to explore and combine measurements to make new findings.

Figure 2: An example of a software dashboard that is trying to imitate a physical counterpart (Few, 2006)

2.3 Measuring

This section will define and explain the terms *measurement*, *indicators* and KPI/PI. The purpose is to provide a better understanding of these concepts as they are important aspects of visualization.

2.3.1 Measurements and indicators

According to ISO/IEC 15939 (International Standard Organization and International Electrotechnical Commission, 2007) a measurement process has the purpose of collecting, analyzing and reporting data, with the intention of managing and demonstrating the quality of processes and products. Implementing measurement systems enables a company to improve in a more efficient way, since it has the capability to measure its performance in key areas. A measure in itself is not a usable measurement; it should have a purpose and be something upon which it's possible to base actions (e.g. measured productivity in a company). An *indicator* (e.g. average productivity) is a variable that evaluates the result of a measurement (e.g. productivity of different projects) based on an *analysis model*, and it is the indicator (see Figure 3) that is usually viewed by the user. The *analysis model* is a calculation on one or more measures with the goal of evaluating performed measures (e.g. apply thresholds to show a value as bad/good or red/green).

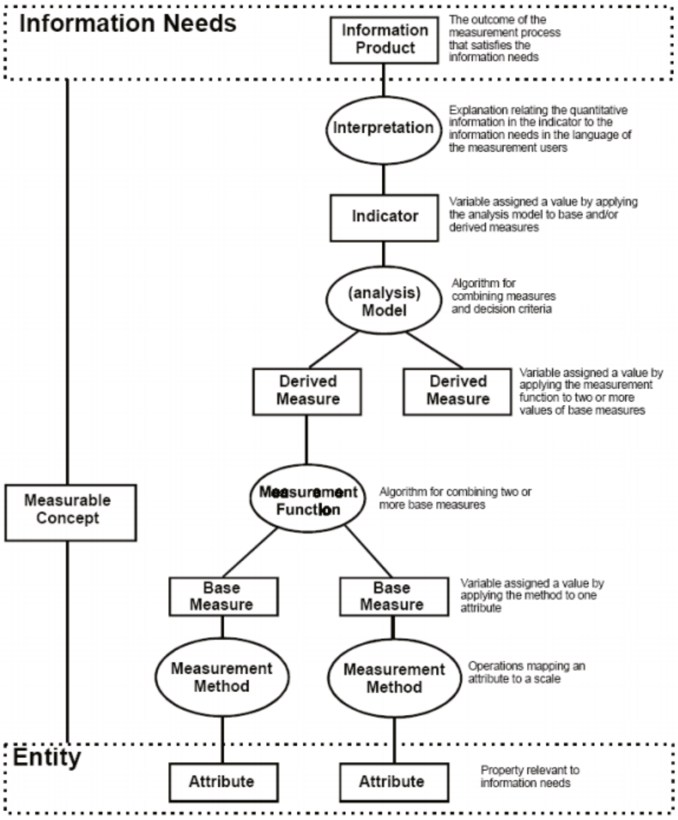

Figure 3: ISO 15939 Measurement Information Model

2.3.2 Developing measurement systems

According to Bourne, Mills, Wilcox, Neely, & Platts (2000), performance measurement systems should be developed in three different development phases (see Figure 4). These phases are; initially designing performance measures, implementation of measures, and using the measures. In the design phase, the question of *what* should be measured should be answered. In the second phase, implementation, the measurement systems are realized. This means that the data of the measurement systems need is properly captured and visualized. The last phase consists of two separate steps. The first is using the measures; it includes activities such as ensuring that the measurements are correct in relation to the strategy of a group or a company. The other step in this phase is a reflection step, during which the measures should be used to contest the currently used strategy.

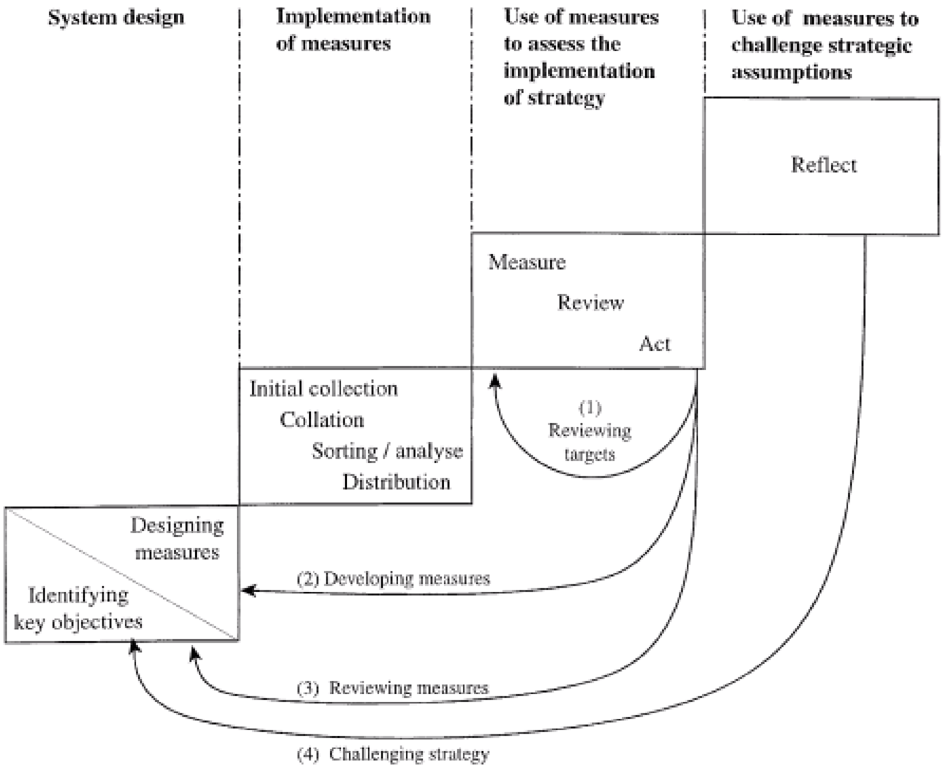

Figure 4: The different steps of creating a measurement system (Bourne, Mills, Wilcox, Neely, & Platts, 2000)

2.3.3 Key performance indicators (KPI)

A KPI (key performance indicator) is a business metric that indicates what is needed to make *major* performance improvements (Parmenter, 2007). It represents several measures which together make out the most important factors to enable such improvements. The KPI by itself is able to indicate what areas that needs attention for the value to improve (e.g. show number of late plane departures, the action should then be to improve plane departure routines).

2.3.4 Performance indicators (PI)

A performance indicator has the purpose of showing the observer what needs to be done in a certain area. It can show an evaluation of a measure which allows the observer to act upon it (Parmenter, 2007), but it is focused on a smaller area than KPIs.

2.3.5 Information quality (IQ)

When showing indicators, it is important that the stakeholder who uses it should be able to act upon it, as well as trust that it is correct. Staron and Meding (2009) have conducted research regarding information quality and reliability. By conducting workshops and interviews they evaluated what participants thought of having an information quality indicator in addition to the normal indicator. This was a step taken in order to ensure the end user that the data presented was indeed correct. They found in their research that showing information quality increases the usage of the measurement tools, since the information quality indicator indicates whether the measured data can be trusted or not (Staron, Meding, & Nilsson, 2009). Staron et al. (2014) have concluded that the persons using the tool have to have mandate to act upon the information given by the indicators, otherwise the displayed data are mostly fancy graphs that can be used for discussions . And when deciding on which indicators to use in tools and measurement systems, research has shown that a high number of indicators are not required, it is more important to have fewer with higher quality of information (Staron, 2012). An example (Feldt, Staron, Hult, & Liljegren, 2013) of creation of an indicator is when an early warning system on error-prone source code files was established by measure its changes, test progress and the outcome of the tests, and changes in the test cases themselves. This was then visualized later in heat maps to provide insight in which files that was most error-prone. Lee et al. (2002) have studied a way to evaluate the current state of IQ in organisations; the research resulted in the AIMQ (AIM quality) methodology. This methodology consists of three components that each on its own help assess the current IQ status in an organisation, as well as provide a ground for identifying gaps in IQ. The AIMQ methodology could be used in this research, IQA could have been used to identify if IQ in this area was possible. The researchers acknowledge the importance of IQ in organisations; however due to time limitations, this research did not include this process.

3 Company context

The company works according to agile and scrum paradigm, having teams that have a holistic responsibility for the features they develop. This has led to an increased demand for measures, customized for the specific needs of each team. The metrics team is responsible for among other, developing and maintaining products for visualizing measures.

One of the earliest data visualizations used MS Excel and MS PowerPoint files to produce a presentation of measurements (see Figure 5). This was a time-consuming process since it required manual calculation and compilation of values. The next step consisted of using Excel-files combined with Visual Basic code that automatically updated the excel files, eliminating the need to create PowerPoint presentations. For example an excel cell could be automatically color-coded in order to show status for a certain data range, as well as the information quality (trustworthiness) of the measures.

The Excel method was demanding of the user, since finding the desired data required browsing through folders and files. Also, the measures that had the most impact (e.g. KPI values) could be mixed with measures of less relevance. The next step in this process was to deliver to the user what the user needed to see, without any major effort needed from the user. This became realized in the form of “gadgets”, a form of application that could display information instantly to the user, without the need to find specific files. The negative aspect of this was that the data was tailored only for leader figures (team leaders etc.) and the data was often not real time (a day old or more).

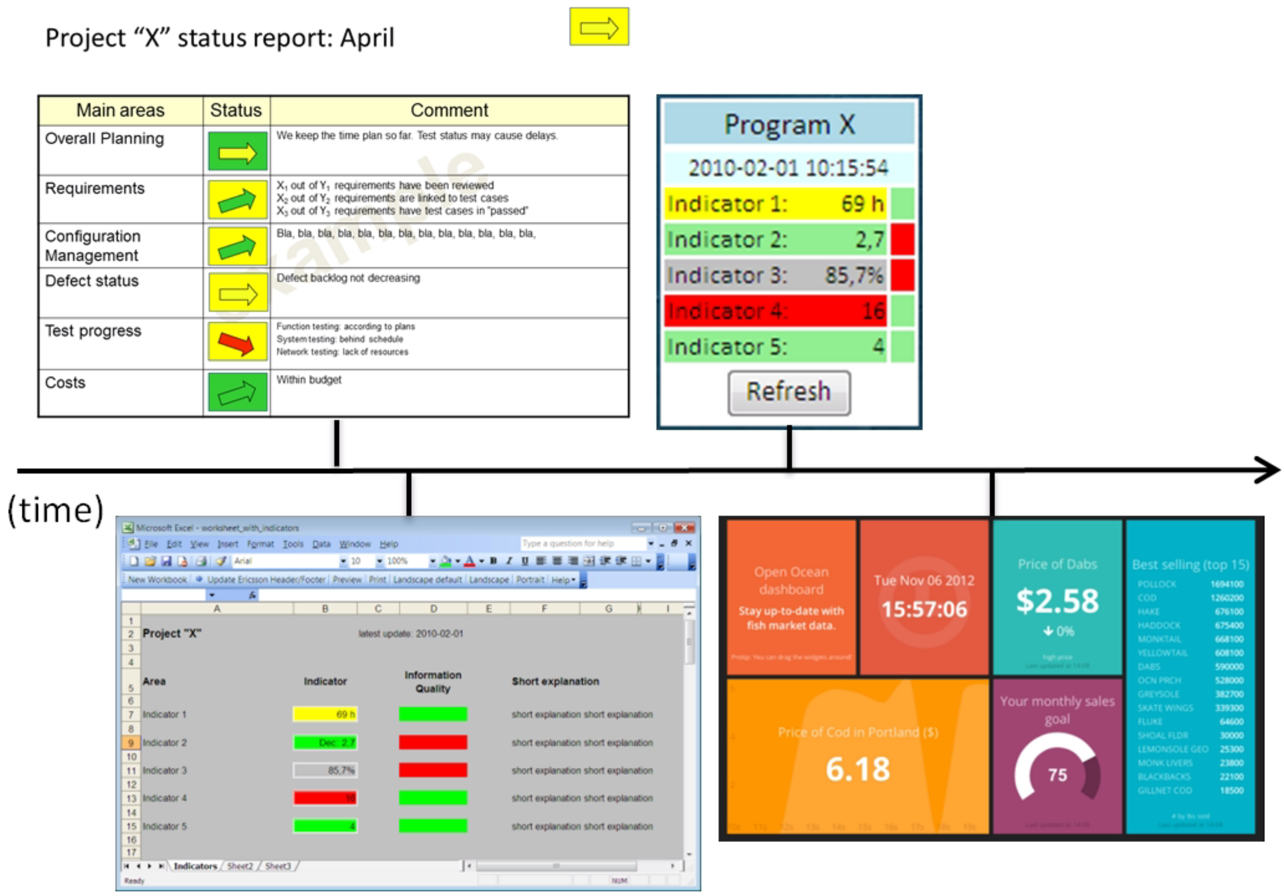

Figure 5: Timeline with examples of types of data visualization, from left to right; Powerpoint, Excel, Gadget, Dashboard.

By the time the company had employed a more team-centred way of working, the need for visualization that accommodated specific team needs was discovered (Staron & Meding, 2014). The teams needed to be able to configure visualizations that were relevant to the specific team's objectives, as well as being relevant to the whole team and not only to the leader. At this point in time, dashboards were chosen as an applicable solution.

There were many different dashboards and information gateways present at the company, developed for specific purposes. These display KPI/PI values that are of relevance to teams (see Figure 6); however this information did not contain everything that was needed to be useful to diagnose the network. Therefore, the company wished to extend their visualization capability by providing dashboards to monitor the current *status* of the network. The *status,* as defined by the company is metrics and trends regarding how many users are currently connected to the network, how much traffic is flowing through the nodes, how many failures have occurred, etc.

The metrics team also had a need of an updated manual and instructions to facilitate further work with dashboards.

Figure 6: Snapshot of a type of network status visualization

4 Related work

There are three subfields of data visualization (Telea, 2014); the first is scientific visualization which focuses on spatial data, such as fluid flow, etc. to give insight into scientific simulations.
The second is information visualization which provides insight into more abstract data, such as documents, source code, and networks. Information visualization unlike scientific visualization gives spatial representations while drawing up the information, rather than the elements providing their spatial placement. All information visualization applications try to answer one question: *"How can we assist users in understanding all that abstract data?"* –Telea (2014)
The last subfield, visual analytics (Cook & Thomas, 2005) (Wong & Thomas, 2004) (Meyer, et al., 2010), is considered as a bridge between the two latter mentioned subfields, but can also be seen as an extension (Telea, 2014). Visual analytics relies heavily on interaction to support different viewpoints to support the observers' sensemaking process, and focus lies on using large datasets and data that is too complex to be viewed in a single display or image (Telea, 2014). This study makes use of both information visualization and visual analytics in order to answer the research question.

This thesis uses some of the strategies and concepts mentioned by Few (2006) for developing dashboards. For example, a description of the importance of data arrangement in dashboards, how the positions should be used for the most relevant visualizations, and how it is important to avoid drawing attention to information that is **not** immediately important. The book also highlights that visualizations which require attention should somehow stand out from the visualizations that tell the observer that everything is in order. In relation to this, it talks about the significance of not obscuring visualizations among unnecessary decorations on a dashboard, rather having a clean interface. Other concepts from the book that the research uses is the notion that a dashboard should be concentrated to an immediate single screen, without the need for scrolling, and that details should be kept to a minimum that enables the observer to understand what is presented, without presenting other irrelevant details regarding the data.

According to Brodbeck and Girardin (2003) it is important to show data in a single frame, and to link the data views to each other to create a sense of cohesion between the data. If there is interactivity, it has to have immediate feedback, and overall the information should be displayed with clarity and with the user experience in mind. The researchers in this thesis used this knowledge to create coherent and clear dashboards. However, the dashboards developed during this research do not adhere to every rule defined by Brodbeck and Girardin (2003).

Burkhard et al. (2005) talks about how important visual representations can be to communicate a message. A human brain is naturally very receptive to motion, shape perception, and colour which is can be utilized when creating visualization. This thesis uses these theories in order to create indicators that communicate a clear message by using colour and animation.

One recommendation from Biehl et al. (2007) regarding dashboards for team situation awareness display, later built upon by Jakobsen et al. (2009), is that when visualizing on a static display for the whole team to see, a dashboard is most effective when it is *glanceable*. By glanceable they mean that first of all the information should be easily interpreted and easily accessed, for example of accessibility: a display all the members of the team can see without leaving their workstation. This research incorporated the design principle of glanceable, although the accessibility part was up to the company to accommodate.

5 Research methodology

This section provides information on the research methodology that this thesis uses.

5.1 Case description

The research was carried out at Ericsson AB. Ericsson is a world leader in communications technology, approximately 40 percent of the world's mobile traffic passes through network equipment supplied by the company.

This study was performed for the metrics team. The metrics team is responsible to create generic metric products for teams and departments. They also have the expert knowledge for where data is located, updated, and extracted.

The researchers wanted to introduce/solve the challenge of visualizing the status of a test telecommunication network, and create a firm foundation to provide a means for the metrics team to be able to generically develop dashboard solutions. The solution will allow agile teams to get dashboards tailored according to their visualization needs. Therefore, as a part of this research, a manual that was to be used as introduction and a guide for members of the metrics team was improved upon.

5.2 Design research

The goal of this research was to find out how the state of a telecommunication network can be visualized. In order to do so, the researchers developed a technical artefact, in the form of dashboards, which visualize selected status indicators of the network. Since the focus of this research would be to develop a new solution, the researchers chose to conduct design research methodology (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007)(see Figure 7). This was preferred over e.g. a case study which would focus on investigating and explaining a current situation rather than focusing on producing a solution (Creswell & Clark, 2009). Von Alan et al (2004) summarize design research as research with the objective of creating an artefact to solve a problem. Peffers et al. (2007) describes the artefact development as one that “should be a search process that draws from existing theories and knowledge to come up with a solution to a defined problem. Finally, the research must be effectively communicated to appropriate audiences”.

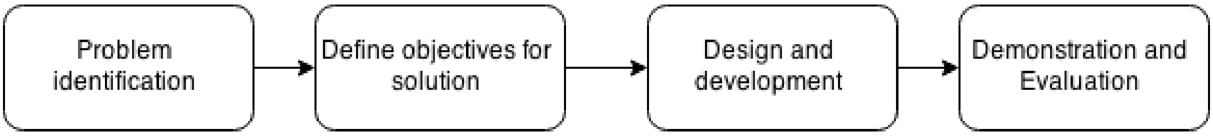

Figure 7: Research steps of this design research

5.2.1 Problem identification

The main research problem was the need to visualize data related to the status of the test telecommunication network. In order to determine how to visualize it, the researchers needed to define *what the network status is*. The *status* for this research was defined as first the “health” of the network. The health concept consists of a selection of indicators for a node, and this information specifically is CPU Load, memory consumption, number of users, dropped data and data flow. A second definition of status is failure ratio of the network. In this context failure ratio is defined as how big the probability of failure is for different operations. To get an understanding of the current state of the challenge, the metrics team provided the researchers with requirements as to what measures they needed to be retrieved and visualized. It also gave the researchers pointers on how it should be presented. Additionally, the researchers studied the current state of measure visualization in the metrics team to get an appreciation as to what techniques currently existed and where improvements could be made. The second challenge that was identified was that the metrics team wanted to facilitate further use of visualization tools. Thus the researchers identified two focus areas; data collection and status visualization.

5.2.2 Defining solution

When the problem to be solved was defined, the researchers studied possible solutions to both the visualization aspect and the data collection aspect. Tools and methods that could help the visualization aspect of the artefact were studied, as well as techniques for properly extracting the desired data from the source database. The researchers were somewhat limited in their choice of tools, as the company had already used certain tools in similar situations and wanted to continue using them. In this stage of the research, the researchers learned how to use the tools that were in place and the tools that were needed to solve the problem according to the problem identification phase.

The solution that was defined was that dashboards to visualize the network status would be developed using an open source library called Dashing-js, and the data would be retrieved from a database. The Dashing dashboards were to contain widgets that showed KPIs and PIs. To solve the challenge of facilitating learning, an existing manual was improved. The manual included instructions on how to use Dashing to build widgets and dashboards using a custom data source.

5.2.3 Developing and designing

Once the research on the techniques the company had selected was complete, the development phase started. Here, the gathered research on how and what measures to visualize was conceptualized in a prototype. The prototype was continually evaluated by the team leader and the database expert. By taking their input, the researchers could adapt the prototype to solve the company's visualization need. The prototype took the form of several dashboards representing different areas of measurements (e.g. health, fail ratio) that automatically rotated between them. Alongside this development, a manual was revised with better instructions on how to develop dashboards and widgets.

5.2.4 Demonstration and evaluation

The prototype was tested and verified using live data provided by the company. Three employees from the company: the team leader for the metrics team, the product owner for the test telecommunication network, and the measurement expert were shown the product and could verify that it did indeed solve the problem. As the last step of evaluation the researchers developed an alternative dashboard to compare against the developed prototype.

The interviewees were selected since they were stakeholders who in their different roles had interest in visualizations of the telecommunication network using dashboards. By interviewing a sample of different roles, the researchers could understand how well the visualizations met the interviewees' requirements.

- The team leader was selected since he was the supervisor of the research and had interest in the resulting visualization, and was also responsible for maintaining and developing future dashboards in the company and to ensure they were easy to use (see section 4).
- The product owner was selected because of his knowledge and interest in the test telecommunication network and the need of visualizations for it. The product owner had knowledge about what type of displays were needed and what data should be presented (see section 2.3.5).
- The database administrator/designer was selected since he had the ability to determine if the right data was visualized and if it was in a correct way (see section 2.3.5).

The interviews were conducted in a semi-structured way, allowing the interviewees to express their opinions on the different dashboard alternatives. The interview provided a quantitative score motivated by a qualitative answer in order for the researchers to determine what the positive and negative aspects of the dashboard alternatives were. The questions were formulated in a way that let the researchers know if the visualization of the network was good, and also gave them information on what could be improved in future work. The answers to the questions also showed what the interviewees considered when viewing the dashboards. By using a scale from 1-10, a score was produced that facilitated the comparison between the alternatives. The scores helped the researchers conclude which visualization subfield (see section 4) that was the most effective, if any. Since the answers told the researchers how well the visualizations worked, and what was missing, they were able to conclude if it's effective to visualize the status of telecommunication networks using dashboards and how to do it most effectively.

Interview questions:

- What role do you have in the organization?
- On a scale of 1 – 10, 1 being bad and 10 being perfect, how good do you think the dashboard is?

    - Motivate your score, what is missing? What is good?
- On a scale of 1 – 10, 1 being bad and 10 being perfect, how good do you think the alternative interactive concept is?
    - Motivate your score, what is missing? What is good?
- On a scale of 1 – 10, 1 being bad and 10 being perfect, how good do you think the alternative non interactive concept is?
    - Motivate your score, what is missing? What is good?
- Which alternative do you think is better? Motivate why.

5.3 Data collection

For each new feature to add to the dashboards, the researchers had to investigate, by coordinating with the database expert of the team, where to locate that particular data set and the data from the set that was of relevance.

5.3.1 Missing data

One of the features the researchers were asked to explore, was to visualize a graph that was to be updated once every second or at least with a few seconds interval. However, the researchers did not have access to the tools necessary.

5.4 Validity discussion

Validity tells the reader about possible threats to the trustworthiness of the research (Runeson & Höst, 2009). The threats in this thesis are categorized into *internal* which are defined as threats due to the researcher's lack of awareness of influencing factors, and *external* which are defined as threats to generalizability of the findings.

Internal – Only used one tool to gather information, however there might be other tools that provide more or other kinds of information. The amount of time during which the research took place did not allow the researchers to study all of the available tools.

The dashboards where developed as a product of this research, however the data displayed is foreign to the researches. Therefore a threat exists that the researchers might have selected the wrong data due to misunderstanding what the data represent. To minimize this risk, experts from the metrics team have been asked to look over the presented data by the extent of their time available.

Interviews were only conducted with three employees, which can weaken the validation. However the limited amount of time available for research did not allow for further interviews.

The interviewees were already familiar with Dashing dashboards when the interviews were conducted, which might induce bias. The interviewees might already carry an opinion on what they want to see improved in Dashing dashboards, and therefore find the alternatives more appealing.

External – The results of this thesis are specific to the studied company. The researchers believe the results can be generalized, but due to limited time they have not verified this assumption.

6 Results

6.1 Exploratory development

When the researches started this research the company had already initiated work of creating visualizations using dashboards. The existing dashboards that were present at the company were created with the open source project Dashing (Shopify, 2012), or rather an abbreviation called Dashing-js (Caseri, 2013). Dashing-js was already installed and configured on the company servers, and the researchers were required to use the existing framework, i.e. Dashing-js. In order to learn Dashing dashboards and the way of working with jobs, exploration of the existing Dashing dashboard (see Figure 8) was the first logical step. Here the researchers corrected minor bugs such as a counter that separated values into two different widgets that did the separation wrongly and thus the dashboard displayed wrong information.

Figure 8: Initial dashboard

Once the researchers had knowledge about how to create both dashboards and widgets, creating new and editing already existing, exploration of the base-widgets that yielded in a mapping of nine different widgets and their functionality. Having fundamental knowledge increased the development speed since the researchers then just had to extend the existing widgets to get their functionality and then only have to include the new desired improvements or customizations. At the same time a list of the six included libraries was compiled (see Table 2) in order for the researchers to gain better understanding of what kind of widgets that could be developed without including further libraries; from this a widget representing a multiple area graph was constructed (see Figure 12).

The metrics team had an existing manual on how to develop with Dashing, although not complete. Therefore that manual was updated and improved along the way as the research progressed.

6.2 Dashboards with Dashing-js

Dashing is written in Ruby and based on the Sinatra framework (Mizerany, 2007), but Dashing-js is written in JavaScript with Node.js (Joyent, Inc., 2009). The Dashing dashboards can be customized in terms of the widgets they provide; a widget can contain all kinds of visualizations as long as it can be designed in HTML. Each widget consists of three parts (see Figure 9), a HTML part which controls the basic layout and the names of those parts which later can be accessed by the corresponding

CoffeeScript (Ashkenas, 2009) . The CoffeeScript controls the behaviour of the widget depending on what data is received. The final part is Sass (Hampton, Weizenbaum, & Eppstein, 2006) that compiles into CSS which is the look and feel of the widget. Dashing-js comes bundled with nine pre-made widgets (see Table 1).

Name	**Function**
Text	Displays text message
Graph	Displays graph
List	Displays list of labels and values
Clock	Displays the time
Meter	Displays a gauge
Comments	Iterates through a list of sentences and images
Number	Displays a number
Image	Displays an image
Iframe	Allows the user to include a HTML page

Table 1: Widget included in Dashing-js

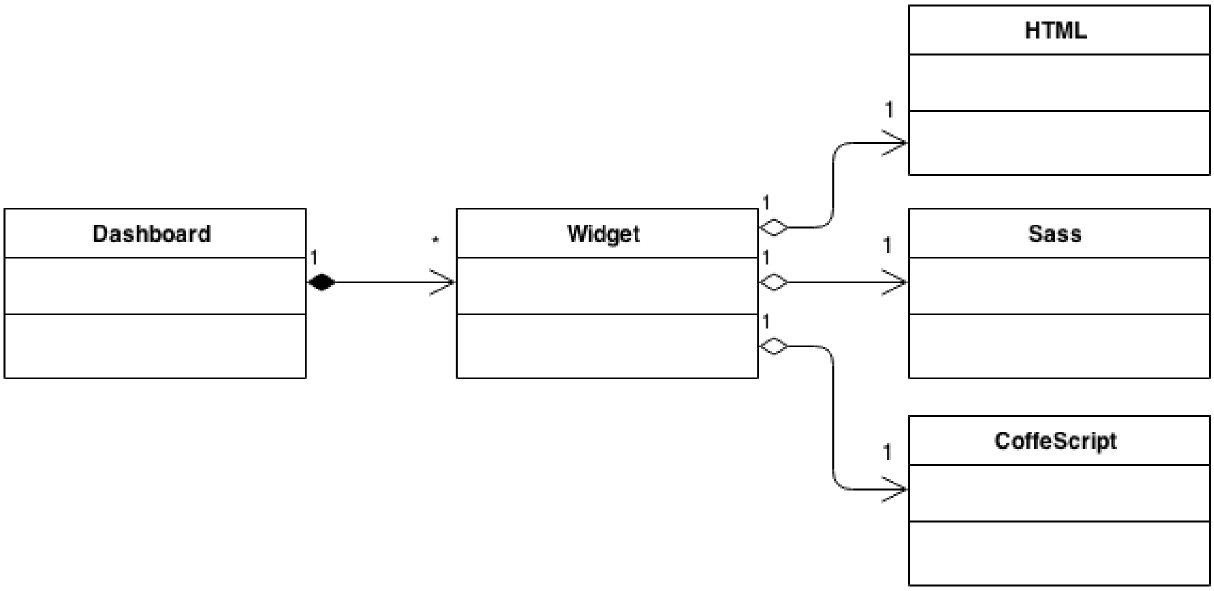

Figure 9: Basic view of Dashing dashboards

A separate part of the Dashing dashboard are jobs; active in the background. These jobs are written in JavaScript but with the extended abilities from Node.js which allows developers to access databases and other data sources. These jobs communicate with the different widgets through their CoffeeScripts.

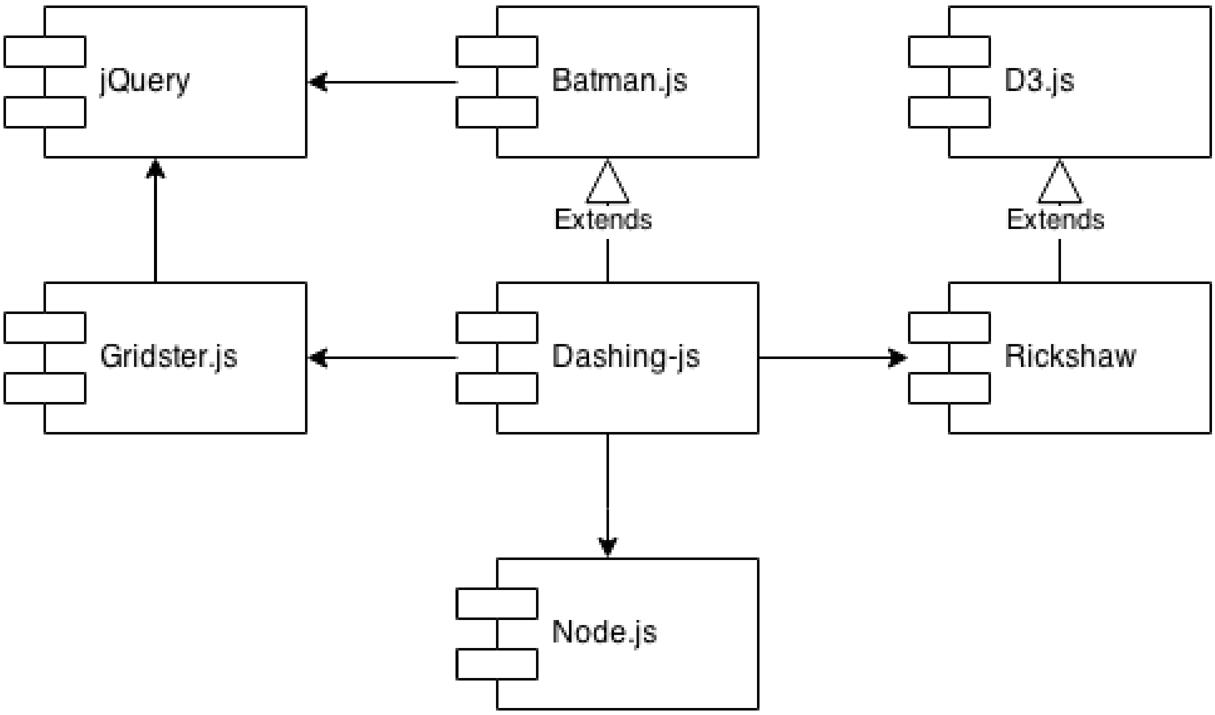

Figure 10: The different libraries composing dashing-js

Dashing is bundled with JavaScript libraries (see Figure 10) to provide functionality out of the box. These libraries include jQuery (The jQuery Foundation, 2006), Batman.js (Shopify, 2011), D3.js (Bostock, 2010), and Rickshaw (Shutterstock, Inc., 2011). The latter is an extension of D3.js (Bostock, 2010) which allows for simple creation of different graphs (see Table 2). Dashing also uses grindster.js (Ducksboard, 2012), a JavaScript library for positioning elements and easily re-arranges them by drag and drop actions.

Library	**Description**
Node.js	Platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. Contains server functionality. (Joyent, Inc., 2009)
Batman.js	Framework for building single page applications. Contains functions for dynamically updating web page values. (Shopify, 2011)
jQuery	Fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. Contains functions for manipulating web page content. (The jQuery Foundation, 2006)
D3.js	JavaScript library for manipulating documents based on data. Helps bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives the full capabilities of modern browsers without tying to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation. Contains functions for drawing graphs and diagrams. (Bostock, 2010)
Rickshaw	Rickshaw provides the elements needed to create interactive graphs. Extends D3.js functionality. (Shutterstock, Inc., 2011)
Gridster.js	jQuery plugin that allows building intuitive draggable layouts from elements

	spanning multiple columns. Contains functions that enable a dynamic layout. (Ducksboard, 2012)

Table 2: Libraries included in Dashing-js

6.3 Deliverables

The dashboards can be divided into two categories; end-user information, i.e. what information end-users might find interesting. And the second category is customer information, i.e. more high level information. The research also provides a platform for further development of dashboards, consisting of commented code and an improved manual on how to create and maintain dashboards.

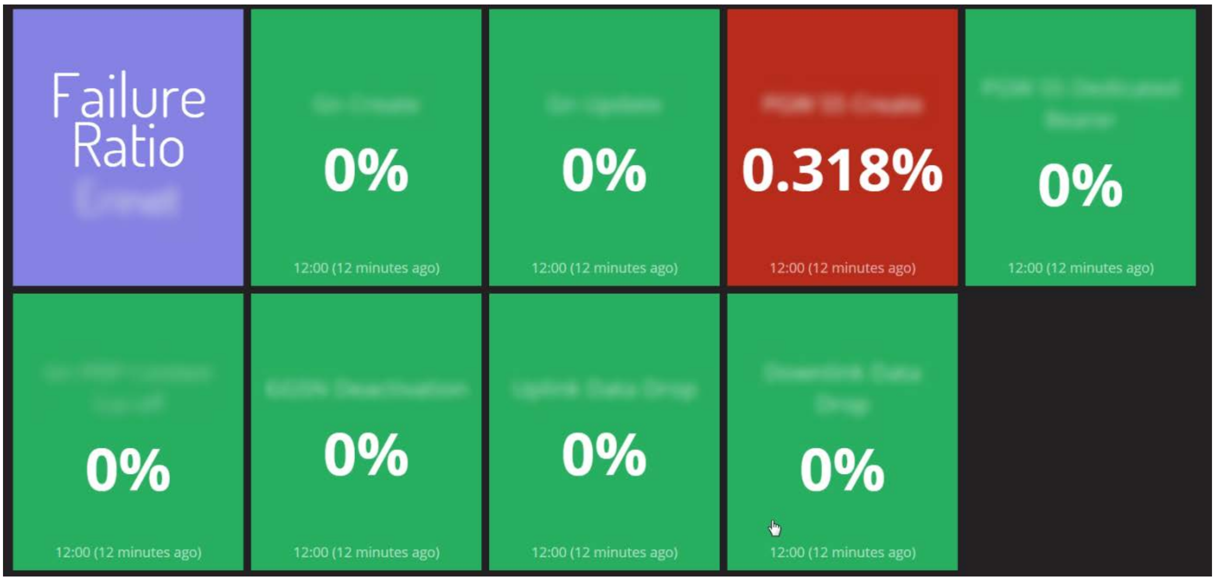

Figure 11: A dashboard containing widgets showing the failure ratio, one of them is pulsating red since the value is over a set threshold

In total six dashboards that displayed status of the test telecommunication network were developed; they contained information on the status of the different nodes in the network (see Figure 11). The status included information on the CPU and memory consumption, the number of currently active users, and information about uplink and downlink packages and more. The dashboard overview also provided information on some of the failure ratios on the network. An example of such a measure can be uplink and downlink data drop failure ratio (see Figure 11). With these dashboards, the research successfully covered all the researchers defined aspects of the status of the test network.

The researchers were asked to perform a quick study on how a team can interact with the network. Therefore a separate dashboard was created with a purpose of instructing teams on how to attach to the network and interact with it in real time (see Figure 12). It showed textual instructions on how to connect to the test telecommunication network using Wi-Fi. The intent of this dashboard was that a team could connect to the network, and via the dashboard be able to survey the current bandwidth they had access to on that network. This bandwidth was at later stages meant to be changed by an interactive screen nearby and thus the observer could both see the difference on the graph displayed on the dashboard, as well as experience changes in real time by conducting bandwidth intensive tasks on the device connected to the network (such as streaming video). This way the user could both see and affect the status of the network in real time. This study verified the possibility to interact with the network but was however not implemented into the organisation due to it being out of scope for the research.

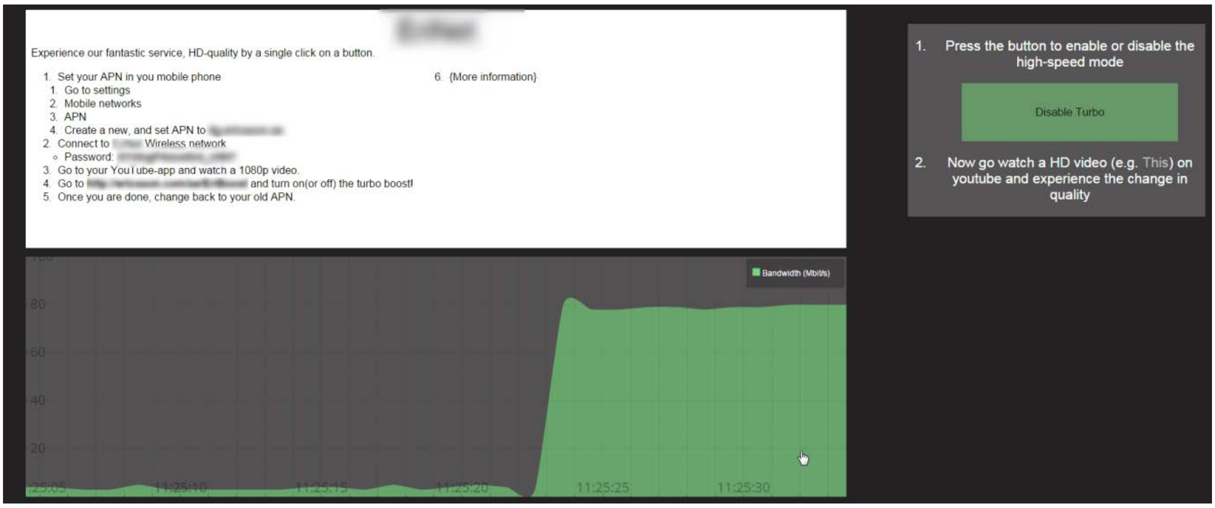

Figure 12: The network bandwidth graph and connection instructions to the left, and the concept turbo (bandwidth increase) enabler to the right, this turbo enabler would preferably be presented on a mobile device separate from the graph.

When designing the dashboards, certain design aspects were taken into consideration. To indicate the warning levels the researchers used the classic stoplight colour coding (Few, 2006) to indicate whether a value had crossed over a certain threshold and required attention, or if it was within predefined limits. For example, if an indicator/widget received a value higher than a set warning limit, it would pulsate in a yellow colour. It did this with the intent to quickly signal that the value needed attention, without the observer to actually needing to read the value first, thus utilizing the human brains attention to motion (Burkhard, Spescha, & Meier, 2005). When a widget received a value that was above a certain threshold, it pulsated red in order to indicate that the value was not only a risk, but also posed a risk to the functionality of the system. By applying the pulsating animation to the values, the researchers somewhat circumvented the issue of people with deuteranopia (red-green colour blindness) being unable to spot a value that was above the warning threshold. The researchers also grouped the widgets in a sensible way. For example, the widgets related to ratios were all grouped together in a dashboard, and the data concerning the network configuration data were grouped together. The researchers grouped these partially by common sense, but also based on input by experts in the metrics team who were knowledgeable as to what data would fit together, and maybe most importantly *who* the audience for the dashboard was. The dashboards themselves were also structured in a way that made sense, i.e. widgets that correlated to each other were grouped above each other. For example the widget for number of packets transferred was placed above the cpu usage, since they affect each other.

One contribution for this research was the researchers work on updating and adding to the pre-existing manual on how to work with Dashing. This manual was initially created before this research project began by an employee from the team. The manual describes how to start with dashing, what the dashing framework consists of, how to create widgets, and additional details that are necessary to fully utilize the dashing framework in the context of network visualization at the company.

Another contribution was an extension on an already existing job that wrote to a Microsoft Excel sheet that utilized four of the widgets placed on the pre-existing dashboard (see Figure 8) that kept track of execution of scripts. The purpose of the Excel sheet was to store and give access for the metrics team leader to observe and track trends of successful script executions, also to incorporate strategic goals such as a certain amount of failed executions per year. This is a good example of a complete cycle on how measures are derived, analysed, visualized, and then stored to track trends and make use of all the measures.

7 Comparative evaluation

In order to evaluate the results, the researchers developed an alternative dashboard of how the data can be visualized. This dashboard was developed using JavaScript following the theories of visual analytics; with the intent of being more interactive and providing a higher overview, rather than the static Dashing dashboards following the theories of information visualization, the "JavaScript" dashboard was therefore divided into two separate parts (see Figure 13 and Figure 14).

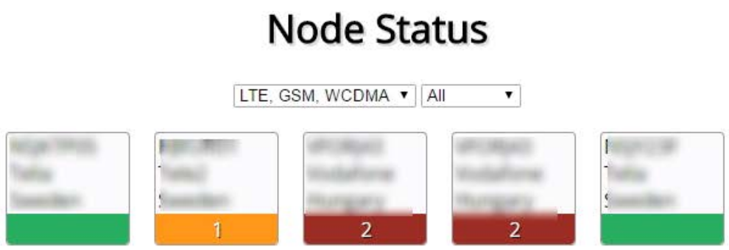

Figure 13: Overview of nodes with the "JavaScript" dashboard

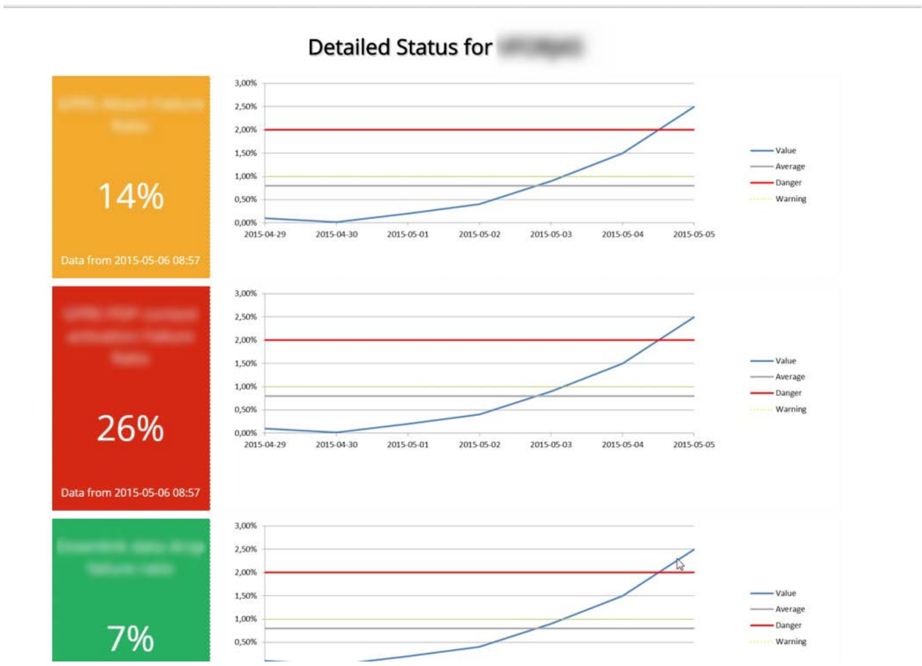

Figure 14: In-depth node information

7.1 Description of "JavaScript" dashboard

The first part of the dashboard was an overview of all the nodes in the displayed network (see Figure 13), here the observer could easily get a quick look to see if there was anything wrong, and also filter which nodes to view by using dropdown menus. The nodes with either warnings or failures were colour coded accordingly, as well as displayed the total number of warnings and failures for that node. The observer could then hover with the mouse over a specific node and see failures, warnings, and a comment on why this is happening (see Figure 15).

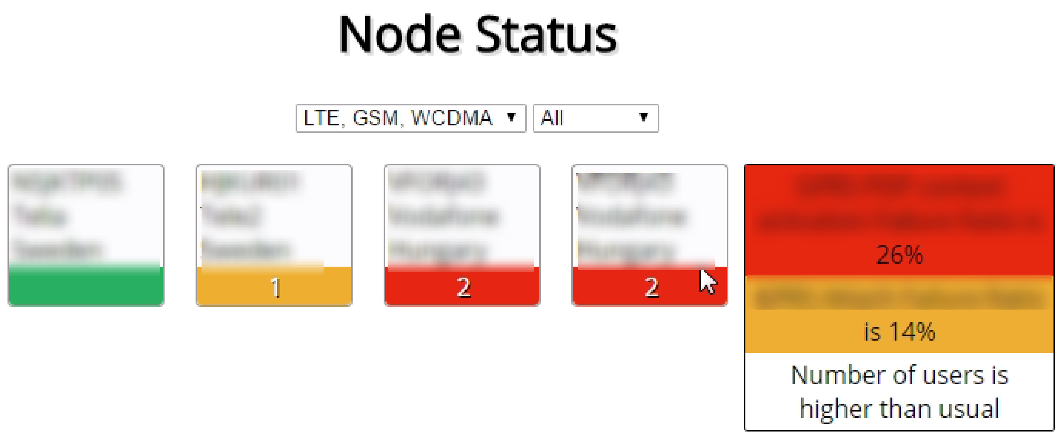

Figure 15: Mouse over menu

The second part was accessible by clicking on a node, displaying more in-depth information about the node (see Figure 14). Here, all the measures for the node are displayed with one real-time value each, and an accompanying graph showing the trends for this measure over time; the graph also shows the values for median, warning, and failure (see Figure 16).

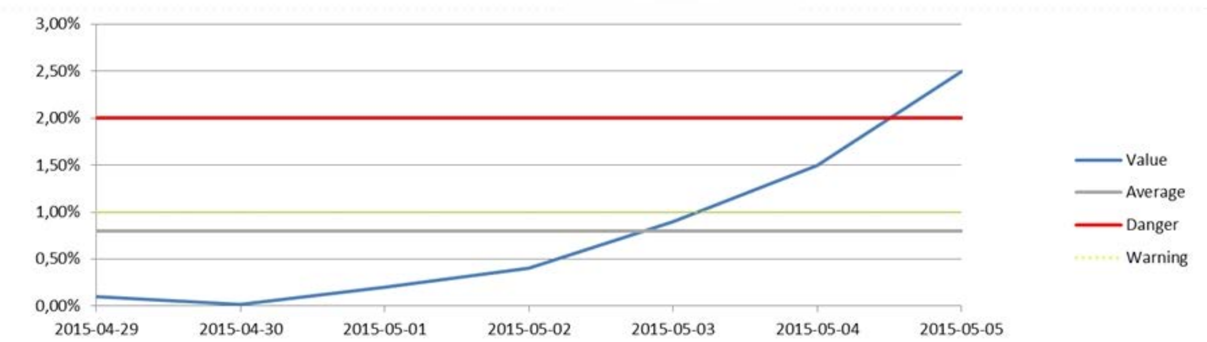

Figure 16: Graph showing values over time

7.2 Static dashboard

A static version of the "JavaScript" dashboard was also created using screenshots placed in an MS PowerPoint file; which rotated between the different nodes and after each one it showed the node overview, the use of MS PowerPoint was to increase development speed instead of actually implementing a rotation feature. The rationale for creating the static version of the "JavaScript" dashboard was to have a version that could be used on TV-screens.

7.3 Comparison of Dashing dashboard, "JavaScript" dashboard, and Static dashboard

To compare the three different approaches: Dashing dashboard, "JavaScript" dashboard, and Static dashboard, interviews were held with three employees:

- Team leader for the metrics team

- Product owner for the test network
- Database administrator (DBA) and designer

The field *Time spent on creation* is based on a rough estimate; however the Dashing dashboard is a fully functional dashboard that visualizes the measures of a test network. Whereas the alternative concept consists of dummy data and would need additional time for it to be functional.

	Dashing dashboard	**"JavaScript" dashboard**	**Static dashboard**
General information			
Interactive	No	Yes	No
Framework-based	Yes	No	No
Drilldown	No	Yes	Yes
Colour notification	Yes	Yes	Yes
Shows trends	No	Yes	Yes
Highest overview	Rotates between one node at the time.	Shows all nodes and the number for warnings/failures on one page.	Shows all nodes and the number for warnings/failures on one page. Also rotates between one node at the time.
Visualization sub-field	Information Visualization (see section 4)	Visual Analytics (see section 4)	*Not applicable*
Time spent on development	5 hours	15 hours	15 hours
Score (1-10)			
Metrics Team Leader	10	10	5
Product Owner for the Test Network	2	8	4
DBA and Designer	10	7	7
Comments			
Metrics Team Leader	Just want to know if there are failures, therefore a high-level website is exactly what we need.	Very good for newcomers to the team, gives good understanding and a structural overview. The drilldown functionality is very good.	Useless for the metrics team. This visualization contains none of the aspects that make the others good, since it's non interactive but has worse visualization than the dashboard alternative.
Product Owner for the Test Network	Good for newcomers. But lacks history/trends for the displayed measures, which is a must for understanding what one is looking at.	Generally better than the first [Dashing]. Good with drilldown, however the graphs should also be interactive, the user could then specify what to view in more detail. When viewing real-time data there	Prefer the one with interactivity. This would not show enough information.

		are some data loss and it would be better to see a median value over a certain time.	
DBA and Designer	Shows the big picture, perfect for overview. Good color-coding. Miss a "should be" value for all the measures. Miss a drilldown feature.	Good for my role to see if something isn't working. Would like to combine this with the first [Dashing] dashboard.	This alternative does not add any value over the other two alternatives.
Most preferred			
Metrics Team Leader		X	
Product Owner for the Test Network		X	
DBA and Designer	X		

Table 3: Summary of interviews

The interviewees were shown one alternative at a time, and was allowed time to get a perception of how it worked. The interviewee was then asked about a score for the alternative and a motivation for that score. The interviewee was finally asked to select one alternative that was better than the other two and asked to motivate their choice.

In general, the interviewees thought the Dashing dashboard and the "JavaScript" dashboard were the best visualizations, and the Static dashboard was the least favoured. This can be related to the fact that Dashing dashboard had all data present in the main view, and therefore fulfilled the rule that states that all the data should be shown in a single view, was glanceable, and had clear unobstructed view of indicators. It also followed the theories of the information visualization subfield. The Dashing dashboard did not however provide interactivity. The difference between the Dashing dashboard and the "JavaScript" dashboard was that the latter did not provide all data in a single view; however it provided responsive interactivity and the ability to select the data of interest, and had a glanceable interface. It also followed the theories of the visual analytics subfield. The Static dashboard however, did neither provide a glanceable interface, nor the single frame view, nor interactivity. It also didn't follow the principles of any visualization subfield. Therefore it's possible to see a relation between the rules set up by Brodbeck and Girardin (2003), the visualization subfields defined by Telea (2014) and the theories on the importance of glanceable interfaces by Jakobsen et al. (2009), and the score each alternative received.

From these interviews it becomes apparent that drilldown functionality is requested and appreciated by all interviewees. This was the motivation for "JavaScript" dashboard having the highest score and was therefore the most preferred alternative. Other requests are to provide more information on the Dashing dashboards in order to optimize it for further use, such as what units the data is in, and the "should-be" values (what value is acceptable). Real-time data should not be used unless the input is of high frequency, instead a median value over a certain period of time would be more valuable.

During the interviews the researchers did not take into account or mention the development time for the different alternatives. This was due to the fact that the researchers in this interview wanted to focus mainly on the visualization aspects of the different alternatives, and how well they met the needs of the interviewee.

7.4 Speed of development with Dashing

The development time differs greatly between the different products, and can be a factor for choosing a visualization solution. For example, the Dashing dashboard alternative could in theory be developed relatively quickly compared to the alternatives, considering many graphs and widgets are pre-made and only need to be fed the right data in order to visualize them. Meanwhile the "JavaScript" dashboard which the researchers developed would need a customized visualization solution to meet the needs of the team members. An assumption is that development time would increase.

One example of quick development time with Dashing is the development of the dashboard which instructed stakeholders how to connect to the test network. This dashboard also allowed them to see the quality of service with a line graph (see Figure 12). Since the researchers already had the widget for a line graph, creation took 20 to 30 minutes. Most of the development time was spent on creating the Dashing job which supplied the dashboard with data. This can be compared to creating a dashboard from the ground up as with the "JavaScript" dashboard, which even as a concept took several hours to complete.

7.5 Information quality

As mentioned in the background section, information quality (IQ) is an important aspect of a measurement system, as well as when displaying the measures. The researchers wanted to implement IQ functionality in the jobs that deliver data, so that the observer could determine the IQ of the visualization. However, as with all research projects, time had to be prioritized and therefore it was decided that there was too little time to implement such a feature.

8 Conclusion

The goal of this thesis was to research a way to visualize the status of a telecommunication network using dashboards, and to promote further dashboard development at the metrics team. This was realized using the Dashing framework to create dashboards with indicators of the status of a test network and field networks.

Validation was performed by conducting interviews with three employees; a team leader, a product owner, and a database administrator / designer. The employees compared the dashboards using Dashing with an alternative visualization concept. The result of the interviews showed that the Dashing dashboards provided a desired functionality; however the most preferred alternative was the "JavaScript" dashboard due to the fact of it incorporating interactivity, and therefore visual analytics should be considered during development of dashboards. All interviewed employees were positive to the drilldown functionality provided in the alternative concept.

Based on the results, the research shows that developing visualizations using dashboards does provide a good overview of the telecommunication network status. The developed dashboards are compliant with design recommendations, such as using stoplight colour coding for warnings, important values standing out from the rest, being glanceable, and a clean and unobscured interface. It also shows that visualising telecommunication networks can best be achieved by using a framework familiar to the developer and based on interviews, incorporating drilldown is important.

From all of these results, the researchers can draw the following recommendations regarding visualizing the status of a telecommunication network:

- Consider following the theories of visual analytics rather than those of information visualization.
- The overview should be simple and be able to be shown on a static display.
- Trends over time and an additional value that tells the observer what normality is for the value observed.
- Interactivity which provides drilldown functionality is important.
- Follow design recommendations and keep the display away from unnecessary clutter.
- Make use of an existing familiar framework; otherwise this research can recommend the Dashing framework based on the researchers experience with it.

For future work, further investigation in the drilldown functionality combined with a framework such as Dashing-js is of interest. The conclusions drawn from this research show that combining drilldown and dashboards would be an appropriate solution for visualizing the status of telecommunication networks. Another aspect of interest is to incorporate IQ into the dashboards, since IQ awareness is important in organisations (see section 2.3.5).

Acknowledgement

The researchers would like to thank their supervisor Dr. Miroslaw Staron for all the help, support, and constant feedback. Also, they would like to thank Wilhelm Meding for his expertise and guidance. And finally, thanks to Ericsson AB for the opportunity to conduct the research at their company.

References

Ashkenas, J. (2009, 12 13). *CoffeeScript*. Retrieved 04 16, 2015, from CoffeeScript: http://coffeescript.org/

Biehl, J. T., Czerwinski, M., Smith, G., & Robertson, G. G. (2007). FASTDash: A Visual Dashboard for Fostering Awareness in Software Teams. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (ss. 1313-1322). New York, NY, USA: ACM.

Bostock, M. (2010, 09 26). *Data-Driven Documents*. Retrieved 04 16, 2015, from D3.js: http://d3js.org/

Bourne, M., Mills, J., Wilcox, M., Neely, A., & Platts, K. (2000). Designing, implementing and updating performance measurement systems. *International Journal of Operations \& Production Management, 20*(7), 754-771.

Brodbeck, D., & Girardin, L. (2003). Design study: Using multiple coordinated views to analyze geo-referenced high-dimensional datasets. *Coordinated and Multiple Views in Exploratory Visualization, 2003. Proceedings. International Conference on*, (ss. 104-111).

Burkhard, R., Spescha, G., & Meier, M. (2005). "A-ha!": how to visualize strategies with complementary visualizations. *Conference on Visualising and Presenting Indicator Systems*, (ss. 1-9).

Caseri, F. (2013, 04 21). *Dashing-js*. Retrieved 04 16, 2015, from Dashing-js: https://github.com/fabiocaseri/dashing-js

Cook, K. A., & Thomas, J. J. (2005). *Illuminating the path: The research and development agenda for visual analytics.* Tech. rep., Pacific Northwest National Laboratory (PNNL), Richland, WA (US).

Creswell, J. W., & Clark, V. L. (2009). Research Design: Qualitative, Quantitative, and Mixed Methods ApproachesResearch Design: Qualitative, Quantitative, and Mixed Methods Approaches.

Ducksboard. (2012, 07 15). *Gridster.js*. Retrieved 04 16, 2015, from Gridster.js: http://gridster.net/

Feldt, R., Staron, M., Hult, E., & Liljegren, T. (2013). Supporting software decision meetings: Heatmaps for visualising test and code measurements. *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*, (ss. 62-69).

Few, S. (2006). *Information dashboard design.* O'Reilly.

Hampton, C., Weizenbaum, N., & Eppstein, C. (den 25 06 2006). *Sass-lang*. Hämtat från Sass-lang: http://sass-lang.com/ den 13 05 2015

International Standard Organization and International Electrotechnical Commission. (2007). *ISO/IEC 15939 Software engineering - Software measurement process.* Geneva: International Standard Organization / International Electrotechnical Commission.

Jakobsen, M. R., Fernandez, R., Czerwinski, M., Inkpen, K., Kulyk, O., & Robertson, G. G. (2009). WIPDash: Work Item and People Dashboard for Software Development Teams. i *Human-Computer Interaction â€“ INTERACT 2009* (Vol. 5727, ss. 791-804). Springer Berlin Heidelberg.

Johansson, L., Meding, W., & Staron, M. (2007). An Industrial Case Study on Visualization of Dependencies between Software Measurements. *7th Conference on Software Engineering Research and Practice in Sweden, 1:2007:02*, p. 23-33.

Joyent, Inc. (2009, 02 15). *Node.js*. Retrieved 04 16, 2015, from Node.js: https://nodejs.org/

Lee, Y. W., Strong, D. M., Kahn, B. K., & Wang, R. Y. (2002). AIMQ: a methodology for information quality assessment. *Information & management*, 133-146.

McKay, J., & Marshall, P. (2001). The dual imperatives of action research. *Information Technology & People, 14*(1), 46-59.

Meyer, J., Thomas, J., Diehl, S., Fisher, B. D., Keim, D. A., Laidlaw, D. H., et al. (2010). From Visualization to Visually Enabled Reasoning. *Scientific visualization: Advanced concepts, 1*, 227-245.

Mizerany, B. (2007, 09 2). *Sinatra*. Retrieved 04 16, 2016, from Sinatra: http://www.sinatrarb.com/

Parmenter, D. (2007). *Key performance indicators (KPI): developing, implementing, and using winning KPIs.* John Wiley \& Sons.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems, 24*(3), 45-77.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering, 14*(2), 131-164.

Shollo, A., Pandazo, K., Staron, M., & Meding, W. (2010). Presenting Software Metrics Indicators: A Case Study. *Presenting Software Metrics Indicators: A Case Study*.

Shopify. (2011, 01 09). *Batman.js*. Retrieved 04 16, 2015, from Batman.js: http://batmanjs.org/

Shopify. (2012, 07 15). *Dashing*. Retrieved 04 16, 2016, from Dashing: http://dashing.io/

Shutterstock, Inc. (2011, 10 16). *Shutterstock*. Retrieved 04 16, 2015, from Rickshaw: http://code.shutterstock.com/rickshaw/

Staron, M. (2012). Critical role of measures in decision processes: managerial and technical measures in the context of large software development organizations. *Information and Software Technology, 54*(8), 887-899.

Staron, M., & Meding, W. (2011). Monitoring bottlenecks in agile and lean software development projects--A method and its industrial use. i *Product-Focused Software Process Improvement* (ss. 3-16). Springer.

Staron, M., & Meding, W. (2014). MetricsCloud: Scaling-Up Metrics Dissemination in Large Organizations. *Advances in Software Engineering, 2014*.

Staron, M., Hansson, J., Feldt, R., Meding, W., Henriksson, A., Nilsson, S., o.a. (2013). Measuring and Visualizing Code Stability--A Case Study at Three Companies. *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on*, (ss. 191-200).

Staron, M., Meding, W., & Nilsson, C. (2009). A framework for developing measurement systems and its industrial evaluation. *Information and Software Technology, 51*(4), 721-737.

Staron, M., Meding, W., & Palm, K. (2012). Release Readiness Indicator for Mature Agile and Lean Software Development Projects. i C. Wohlin (Red.), *Agile Processes in Software Engineering and Extreme Programming* (Vol. 111, ss. 93-107). Springer Berlin Heidelberg.

Staron, M., Meding, W., Hansson, J., Höglund, C., Niesel, K., & Bergmann, V. (2014). Dashboards for continuous monitoring of quality for software product under development.

Telea, A. C. (2014). *Data visualization: principles and practice.* CRC Press.

The jQuery Foundation. (2006, 03 19). *jQuery*. Retrieved 04 16, 2015, from jQuery: https://jquery.com/

Williams, J., Sochats, K., & Morse, E. (1995). Visualization, volume 30. *Annual Review of Information Science and Technology (ARIST)*, 161-207.

von Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly, 28*(1), 75-105.

Wong, P. C., & Thomas, J. (2004). Visual analytics. *IEEE Computer Graphics and Applications, 24*(5), 20-21.